

CLAIMS

1 1. *(currently amended)* A computer cluster comprising:
2 storage media;
3 a first computer having a first instance of an application program
4 installed, said application program having instructions, said first
5 computer including,
6 volatile memory;
7 processing means
8 for executing instructions of said first instance of said
9 application program so as to modify data stored in said
10 volatile memory,
11 for creating a snapshot of said data while said first
12 instance of said application program is running, said
13 snapshot being stored in said volatile memory, and
14 for, while said first instance of said application
15 continues to modify said data so that it diverges from said
16 snapshot, transferring said snapshot from said volatile
17 memory to said storage media while said first instance of
18 said first instance of said application program is running,
19 and
20 a second computer having a second instance of said application
21 program installed, said second computer including means for
22 accessing said storage media so that said second instance of said
23 application can access said snapshot as stored on said storage media.

1 2. *(original)* A computer cluster as recited in Claim 1 wherein said
2 processing means includes
3 a data processor
4 for executing instructions of said first instance of said
5 application program so as to modify data stored in said memory,
6 and
7 for creating said snapshot of said data while said first
8 instance of said application program is running, said snapshot
9 being stored in said volatile memory, and
10 a transfer processor for transferring said snapshot from said
11 volatile memory to said storage media while said first instance of said
12 first instance of said application program is running.

1 3. *(original)* A computer cluster as recited in Claim 1 further
2 comprising a first cluster daemon running on said first computer for
3 causing said snapshot to be created.

1 4. *(original)* A computer cluster as recited in Claim 1 further a second
2 cluster daemon running on said second computer, said second cluster
3 daemon providing:
4 for detecting a failure that prevents said first instance of said
5 application program from running on said first computer, said
6 failure detector, and
7 for causing, in response to said detecting a failure, said
8 second computer to process said snapshot in accordance with
9 instructions of said second instance of said application program.

1 5. *(original)* A computer cluster as recited in Claim 1 wherein said
2 processing means provides for, in response to a write access of a
3 section of said volatile memory in accordance with instructions of said
4 first instance of said application program, copying data in that section
5 so that one instance of said data originally in that section is modified
6 and the other copy of data originally in that section is not modified.

1 6. *(original)* A computer cluster as recited in Claim 2 wherein said
2 data processing means maintains state data, said snapshot data
3 including at least some of said state data.

1 7. *(currently amended)* A method comprising:
2 executing a first instance of an application program on a first
3 computer of a computer cluster so as to generate a series of memory
4 states;
5 creating a snapshot of one of said states; and
6 at least partially during a state that differs from the state
7 represented in said snapshot, transferring said snapshot to storage
8 media accessible by a second computer of said computer cluster.

1 8. *(original)* A method as recited in Claim 7 further comprising
2 executing a second instance of said application program on a second
3 computer of said computer cluster using said snapshot as a starting
4 state.

1 9. *(original)* A method as recited in Claim 8 further comprising
2 detecting a failure that prevents execution of said first instance of said
3 application program, said detecting occurring after said transferring
4 and before said executing a second instance.

1 10. *(original)* A method as recited in Claim 8 wherein said executing a
2 second instance follows said transferring without an intervening
3 detection of a failure.

1 11. *(original)* A method as recited in Claim 7 wherein said transferring
2 is effected by a data transfer processor not used in executing said first
3 instance of said application.

1 12. *(original)* A method as recited in Claim 7 wherein said executing is
2 effected by a data processor that stores processor state data internally,
3 said snapshot including said processor state data.